

Rede IoT assistida por sistema de aeronaves remotamente pilotadas para apoio em operações de recuperação de desastres

Erick M. Moreira^a, Fábio Luiz Jr, Leandro Moreira^a, Jauvane C.de Oliveira^b e Paulo F. F. Rosa^a

^aInstituto Militar de Engenharia (IME)

Praça General Tibúrcio, 80, 22290-270, Praia Vermelha, Rio de Janeiro, RJ, Brasil.

^bLaboratório Nacional de Computação Científica (LNCC)

*emenezes@ime.eb.br

RESUMO: Plataformas IoT (Internet of Things) são consideradas o futuro das redes de sensores. Sua natureza embarcada facilita a adição de mobilidade e amplia drasticamente sua possibilidade de uso. Especialmente, quando usadas com veículos aéreos, essas redes podem avaliar crises rapidamente. Durante as operações de recuperação de desastres, comunicação em tempo real é essencial para coordenar as atividades. No entanto, esses cenários geralmente são privados de boa infraestrutura de comunicação. Este trabalho propõe um sistema capaz de estabelecer uma rede usando veículos aéreos, que conecta todas as partes interessadas através de uma rede de endereços IPv6. Os experimentos mostraram que um pequeno conjunto de 50 aeronaves equipadas com um dispositivo de comunicação com alcance de 50 metros é capaz de conectar até 24 elementos, espalhados por uma área de 13.000 metros quadrados. Para um conjunto de 150 aeronaves, foi possível atender a uma média de 68 agentes, em uma área de 25.600 metros quadrados.

PALAVRAS-CHAVE: Aeronave remotamente pilotada. ARP. Rede mesh.

ABSTRACT: IoT (Internet of Things) platforms are considered the future of sensor networks. Its embedded nature facilitates the addition of mobility and dramatically broadens the possibility of use for such systems. Especially when shipped on aerial vehicles, those networks can be used to assess crises quickly. During disaster recovery operations, instant communication is of the essence to coordinate activities. However, those scenarios are commonly deprived of good communication infrastructure. This work proposes a system capable of deploying an instant network mounted on UAVs, that connects all interested parts through an IPv6 mesh network. The experiments have shown that a small set of 50 UAVs equipped with a 50-meter range communication device is capable of connecting up to 24 elements spread over an area of 13,000 square meters. For a set of 150 UAVs, it was possible to serve an average of 68 agents in an area of 25,600 square meters.

KEYWORDS: Unmanned aerial vehicle. UAV. Mesh network.

1. Introdução

As aeronaves remotamente pilotadas (ARP) são uma tecnologia inestimável para muitas aplicações, como o gerenciamento de desastres [1-5], principalmente devido à sua flexibilidade e alta empregabilidade. Outras tarefas que podem ser realizadas por um conjunto de múltiplas ARPs são: inspeção [6-9], vigilância [1-2] e resposta a emergências [10-11].

Nesses contextos, a diversidade e complexidade

das demandas impossibilita o projeto de máquinas que exerçam todos esses papéis. A solução tem sido criar robôs mais simples, heterogêneos, limitados, que cooperem uns com os outros em um sistema de aeronaves remotamente pilotadas (SARP) para exceder essas limitações, demonstrando unicidade de vontade (objetivo comum) e praticando o coletivismo, a partir do autoconhecimento e da comunicação com seus vizinhos.

Normalmente, operações como essas têm um

centro de comando e controle (centro C2) e agentes de campo espalhados pela área afetada, conforme ilustrado na **figura 1**. Os agentes são responsáveis por detectar danos e reportar ao centro C2, que decide quais ações eles devem executar para mitigar o problema. A rede de comunicação é um elemento crítico desse cenário, pois é através dela que os eventos são relatados e prontamente respondidos; esse canal também possibilita o acompanhamento operacional e logístico das atividades desenvolvidas pelas equipes.

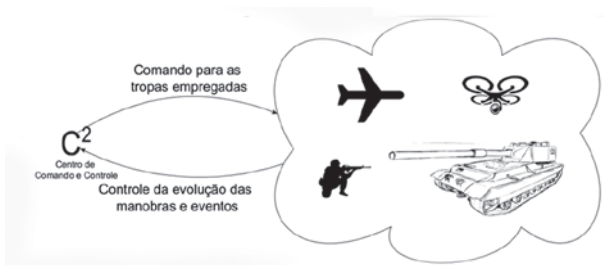


Fig. 1 – O conceito de comando e controle estabelece que o centro C2 deve coordenar as ações de todos os agentes de campo envolvidos [12]. Portanto, ele deve ter consciência situacional durante o maior tempo possível. Isto pode ser alcançado com o recebimento de relatórios dos agentes em campo e através do monitoramento do local da operação.

Um problema fundamental em áreas afetadas por desastres é a ausência ou precariedade de infraestrutura; recursos comuns como o LTE (*Long Term Evolution*, 4G) podem não estar disponíveis. Outras soluções, como caminhões de antenas móveis, postos de comando móveis e torres de celular móveis também podem ser impraticáveis devido às condições das ruas e estradas. Um SARP pode ajudar a melhorar os resultados no cenário descrito. Eles podem monitorar visualmente o local, detectando níveis de radiação, concentrações nocivas de fumaça, ou conectando agentes em solo ao centro C2. A **figura 2** apresenta uma região de desastre, provocado

por chuva ou terremotos e a solução vislumbrada para estabelecer comunicação usando múltiplos ARPs.

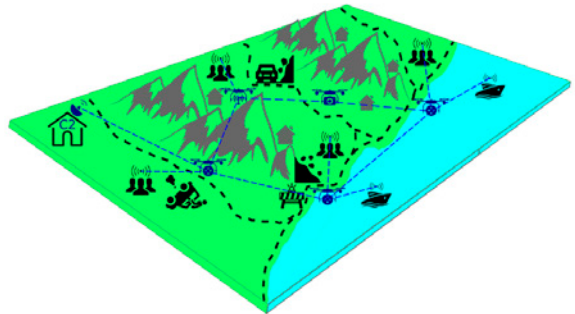


Fig. 2 – A cena mostra eventos desastrosos e equipes de resgate em campo. Devido aos bloqueios de passagem, um SARP foi empregado para estabelecer a conectividade entre os agentes e o centro C2.

Em uma solução como a descrita acima, cada ARP deve ser capaz de estabelecer uma rede de comunicação sem nenhuma infraestrutura prévia. Suas posições devem ser determinadas de acordo com algum critério, e cada veículo deve ser capaz de viajar para sua posição calculada sem colidir com os demais.

Este artigo tem como objetivo apresentar um sistema capaz de calcular essas posições, usando o menor número possível de ARPs. Em seguida, ele deve conduzir cada ARP até a sua posição, evitando colisões. E, finalmente, o sistema deve conectar todos os agentes de campo ao centro C2, através de uma rede mesh estabelecida por dispositivos IoT (*Internet of Things*) usando o software OpenThread [23]. As experiências mostraram que um pequeno conjunto de 50 ARPs, equipados com um dispositivo de comunicação com alcance de 50 metros, é capaz de conectar até 24 elementos, espalhados por uma área de 13.000 m². Para um conjunto de 150 ARPs com as mesmas características, foi possível

atender a uma média de 68 agentes, em uma área de 25.600 m². Já um grupo maior de 250 aeronaves, é capaz de conectar até 96 agentes em uma área de 52.900 m².

O restante do artigo está organizado da seguinte forma: a seção 2 apresenta trabalhos relacionados ao problema e a seção 3 descreve o sistema proposto; os experimentos realizados são apresentados na seção 4 e discutidos na seção 5; finalmente, a seção 6 apresenta as conclusões.

2. Trabalhos relacionados

A maioria dos trabalhos relacionados é limitada a tópicos de pesquisa, como algoritmos de roteamento [13-16] ou de posicionamento para a máxima cobertura [17-20].

É certo que uma rede assistida por SARP tem um grande potencial e ainda representa uma área muito inexplorada [21-22]. Entretanto, existem vários problemas a serem resolvidos até que uma rede desse tipo possa ser considerada estável, equilibrando alcance e largura de banda. Segundo os autores de [21-22], as redes SARP são comumente descritas como redes *ad hoc* com mobilidade, variando de acordo com o tipo de missão. Sua topologia normalmente tem um formato híbrido, mantendo uma estrutura de estrela com a estação de controle de solo (GCS, do inglês ground control station) no centro e uma estrutura de malha (*mesh*) entre as ARPs. Além disso, a topologia não tem estabilidade e pode ser encontrada em um estado estático ou em um estado altamente dinâmico, dependendo da aplicação em que é usada. Outra característica marcante dessa rede é que seus nós perdem e recuperam a conectividade continuamente, seja devido à alta mobilidade, as restrições de energia ou à presença de obstáculos entre os nós.

A abordagem de Abrajano [22] usou redes

tolerantes a atraso, com as ARPs transportando dados entre os locais de desastre e o centro C2. Embora isso seja uma abordagem interessante, essas redes não são de tempo real e seu atraso na entrega das mensagens pode interferir no processo de decisão do centro C2.

Neste trabalho buscamos uma abordagem de rede de topologia *mesh* e totalmente conectada, ou seja, nenhum nó deve se afastar além do alcance de comunicação.

3. Visão geral do sistema

O sistema proposto visa a estabelecer uma rede de comando e controle em uma área afetada por desastres para auxiliar na recuperação da região, atendendo às demandas de tráfego geradas pelas equipes de trabalho em solo. O problema a ser resolvido pode ser formalizado da seguinte forma. Dado um conjunto de demandas de tráfego D e um conjunto de ARPs disponíveis V , calcular as posições a serem ocupadas por cada veículo v , de modo que cada um esteja a uma distância máxima δ dos outros, onde a comunicação entre eles é possível. Quando todos os elementos atingirem suas respectivas posições finais, cada ARP deverá utilizar seu dispositivo IoT para criar uma rede de dados com os outros e aguardar até o final da missão. Para evitar colisões, o sistema pode ser modelado com algumas restrições, tais como planejar as trajetórias em um ambiente livre de obstáculos, e posicionar as ARPs distantes umas das outras no eixo Z (altitude).

A introdução deste sistema é dividida em três partes. A primeira seção descreve os itens essenciais da ARP e o software relacionado. Em seguida, apresentamos o algoritmo usado para calcular as posições de todas as ARPs. Por fim, mostramos o software de rede OpenThread e

como ele foi configurado.

3.1 ARP e sistema de controle

O veículo utilizado no experimento descrito neste artigo foi criado por Ricardo Maroquio Bernardo, pesquisador do Laboratório de Robótica e Inteligência Computacional do IME, é do tipo quadricóptero e foi montado com hardware comercial disponível em prateleira, como mostrado na **figura 3**. O objetivo do projeto é que a ARP seja legalmente habilitada a voar em qualquer lugar, sem restrições, de acordo com a legislação vigente; para tanto, o modelo foi desenhado para ter massa menor que 250 gramas.

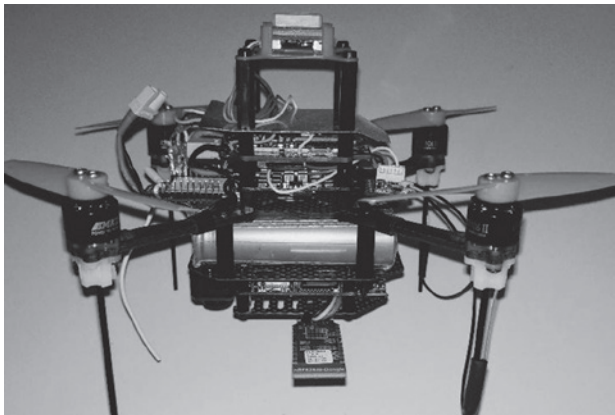


Fig. 3 – Componentes de hardware do quadricóptero: GPS Matek M8Q, controladora de voo Kakute F7 AIO, ESC Luminier 4x1, motores EMax, bateria 2S VTC6 18650 3000 mAh, computador embarcado Nano Pi Neo Air, altímetro do tipo LiDAR, sensor de fluxo óptico e dispositivo de comunicação IoT nRF52840-dongle.

O monitoramento e o controle das ARPs são feitos através de uma GCS (*Mission Planner* [24]). O *firmware* do software ArduCopter [25], na versão 3.7.0, controla a plataforma e é carregado na controladora de vôo Kakute F7 AIO [26], responsável pelo sensoriamento e manobra da aeronave. Junto a essa controladora de vôo, há

um Nano Pi [27], computador embarcado, que executa todos os processos não relacionados ao voo, como o software de rede e outros mecanismos de controle de supervisão distribuídos. A **figura 4** representa o funcionamento do sistema. Todos os processos Linux em execução na ARP têm acesso às variáveis de estado da aeronave, fornecidas pela controladora de vôo, e podem trocar mensagens com o mundo externo através da interface de rede IPv6 provida pelo software *wpantund* [28].

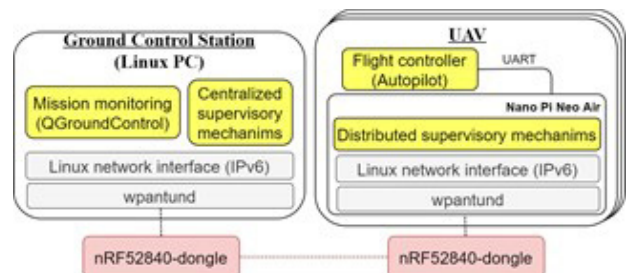


Fig. 4 – A arquitetura de software da plataforma usa o sistema operacional Ubuntu core 16.04 e a aplicação *wpantund* que une o kernel do linux ao dispositivo nRF52840-dongle. Com essa configuração, os processos Linux podem trocar mensagens IPv6/UDP com outros elementos na rede.

Além disso, esse projeto possui uma bateria 2S VTC6 18650 3000 mAh, que permite um tempo de voo de 20 a 25 minutos em boa condição climática.

3.2 Planejamento de posição

No intuito de minimizar a quantidade de veículos utilizadas, desenvolvemos um algoritmo para cálculo das posições finais das aeronaves do SARP.

O algoritmo é baseado no uso do ponto de Fermat ou ponto de Torricelli (pF). Tal ponto determina o local onde a soma das distâncias dos vértices até ele é mínima. O teorema afirma que o ponto existe no interior do triângulo, se nenhum

de seus vértices possui ângulo maior que 120° . Caso haja um vértice com ângulo igual ou maior que 120° , o pF será o próprio vértice. Portanto, teremos algumas demandas de tráfego conectadas diretamente ao centro C2 e outras conectadas por um ponto de Fermat. A primeira é chamada de subárvore direta e a segunda de subárvore de ponto de Fermat (pF). Além disso, cada subárvore tem um custo. O custo de subárvores diretas é a distância entre o centro C2 e a demanda. O custo de uma subárvore de ponto Fermat é a soma de todas as distâncias que conectam o centro C2, ambas as demandas e o ponto Fermat. Os conceitos podem ser vistos na **figura 5**.

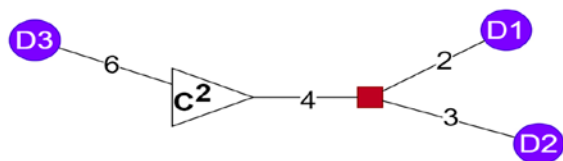


Fig. 5 – A subárvore D1-C²-D2 é do tipo ponto Fermat e têm custo 9, enquanto a subárvore C²-D3 é direta e têm custo 6.

O algoritmo 1 apresenta o planejamento de posição em pseudocódigo. Em síntese, ele consiste em utilizar o máximo de subárvores que atendam 2 demandas por meio de um pF. E, caso alguma demanda não tenha sido atendida, conectá-la diretamente ao centro C2. O resultado é uma lista de subárvores, onde cada uma atende a uma ou duas demandas. Após os cálculos, ainda é necessário percorrer todos os caminhos das subárvores selecionadas para detectar caminhos longos. Se a rota entre dois pontos for maior que a distância estipulada δ , então devemos adicionar entre eles tantas ARPs quanto forem necessários para cumprir a restrição.

Algoritmo 1: Algoritmo de cálculo posições finais	
Dados: conjunto de demandas descritas por sua posição	
Resultado: o conjunto de subárvores de menor custo capaz de conectar todas as demandas ao centro C2	
1	fpTrees \leftarrow Conjunto de todas as subárvores que
2	ligam 2 demandas ao centro C2 através de um
3	ponto de Fermat, em ordem crescente de custo;
4	directTrees \leftarrow Conjunto de todas as subárvores
5	que ligam 1 demanda ao centro C2 diretamente,
6	em ordem crescente de custo;
7	selectedTrees \leftarrow []; visited \leftarrow []; i, j \leftarrow 1;
8	while i < fpTrees.length and j < demands.length
9	do
10	if fpTrees[i].cost \geq demands[j].cost then
11	tree \leftarrow fpTrees[i];
12	i \leftarrow i + 1;
13	demand1 \leftarrow fpTrees[i].demand1;
14	demand2 \leftarrow fpTrees[i].demand2;
15	if demand1, demand2 visited then
16	selectedTrees.add(fpTrees[i]);
17	visited.add(demand1);
18	visited.add(demand2);
19	end
20	else
21	tree \leftarrow demands[j];
22	j \leftarrow j + 1;
23	demand \leftarrow directTrees[i].demand;
24	if demand visited then
25	selectedTrees.add(directTrees[i]);
26	visited.add(demand);
27	end
28	end
29	end

3.3. Rede IoT

Para estabelecer a rede de dados, foi usado o dispositivo nRF52840-dongle fabricado pela Nordic Semiconductor. Seu firmware implementa o software OpenThread no modo de coprocessador de rede (NCP), como mostra a **figura 6**. Como resultado, obtém-se uma interface Linux IPv6 disponível no computador complementar.

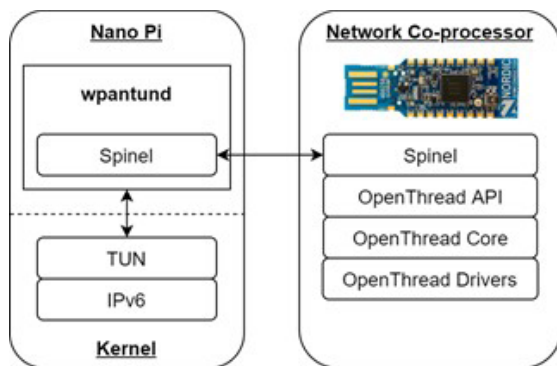


Fig. 6 – Configuração de software e hardware da rede. O computador complementar interage com outros aplicativos por meio de uma interface de rede IPv6, enquanto o wpanund controla o dispositivo NCP usando o protocolo spinel.

OpenThread (OT) é uma implementação de código-fonte aberto da especificação Thread mantida pela empresa Nest Labs. Além da rede mesh formada entre os dispositivos, outro recurso importante disponível na família de software OpenThread é o roteador de borda. Esse elemento é capaz de permitir que outras interfaces de rede entrem na rede mesh, e.g., uma interface sem fio IEEE 802.11bg. É através do roteador de borda que os agentes terrestres podem acessar o restante da rede.

4. Experimentos

As conclusões deste trabalho são derivadas da execução de três experimentos diferentes. O primeiro para determinar o alcance máximo

de comunicação entre duas ARPs e, portanto, determinar o valor máximo de δ a ser usado no cálculo de posições. O segundo experimento determina a quantidade máxima de demandas que podem ser atendidas com diferentes conjuntos de ARPs. E, o terceiro experimento consiste na simulação da missão completa.

No primeiro experimento, movemos progressivamente a ARP para longe da posição inicial, enquanto monitoramos sua localização e o indicador de intensidade do sinal recebido (RSSI). A **figura 7** é um registro do experimento.

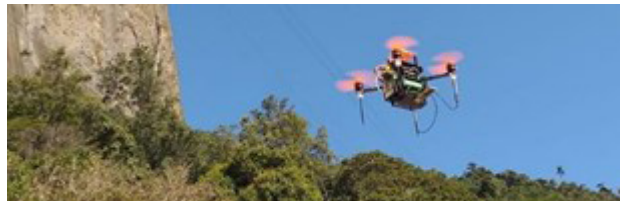


Fig. 7 – O primeiro experimento analisou o alcance da comunicação com o dongle nRF52840 e a taxa de dados disponível.

O objetivo do segundo experimento é determinar quantas demandas podem ser atendidas. Um fator essencial a se considerar é o número de veículos disponíveis e o tamanho da área a ser coberta. Os parâmetros utilizados foram inspirados por [29], que projetou os três cenários descritos na **tabela 1**.

Tab. 1 – Cenários operacionais [29].

Requisito	Cenário 1	Cenário 2	Cenário 3
Área de emprego (m ²)	13.225	25.600	52.900
Número de ARPs disponíveis	50	150	250

Ele propõe três cenários que consideram operações de enxame pequenas, médias e grandes, definindo para cada um a quantidade de ARPs disponíveis e o tamanho das áreas de operação. Nesta fase, foi utilizado o Matlab para determinar a relação entre o número de demandas

a serem atendidas e a quantidade de aeronaves necessárias, usando o algoritmo 1.

O terceiro experimento foi projetado para ser extremamente próximo ao voo real, permitindo avaliar detalhes de um voo com vários robôs. Por esse motivo, usamos a estação de controle de solo (GCS) desenvolvida por nosso grupo de pesquisa e validada para o controle de voos reais com múltiplas aeronaves [30-31].

5. Resultados

O primeiro experimento mostrou que o dispositivo nRF52840-dongle, com antena embutida no circuito, pode atingir até 80 metros com comunicação bidirecional ativa. A **figura 8** mostra um gráfico dos registros do experimento. No eixo horizontal listamos a distância, em metros, e no eixo vertical o nível de sinal RSSI recebido, do inglês Received Signal Strength Indicator. Conforme especificação, o dispositivo possui sensibilidade de recepção máxima de -90 dBm, ou seja, níveis de sinal menores que este indicam que a mensagem não poderá ser recebida corretamente. O limiar é atingido entre 60 e 70 metros, onde há uma queda mais significativa na qualidade da comunicação. Nesse intervalo, aproximamos e afastamos o veículo algumas vezes; por isso, a **figura 8** mostra um gráfico mais caótico nesse intervalo, consequência desses movimentos.

Como toda especificação de rede, a Thread disponibiliza protocolos para resolver problemas comumente apresentados, tais como o protocolo IPv6 para resolver o endereçamento de nós; o protocolo UDP (*User Datagram Protocol*), otimizado para realizar transporte de dados fim-a-fim, aproveitando o cabeçalho de outros protocolos de camadas inferiores; e o protocolo CoAP (*Constrained Application Protocol*), para estabelecer os comandos disponíveis e estruturar

as mensagens no nível de aplicação, em redes com restrição de banda e potência.

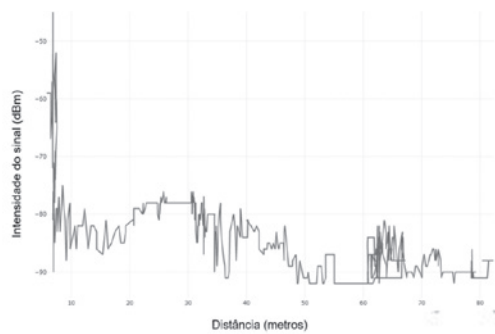


Fig. 8 – Gráfico da intensidade do sinal recebido em função da distância entre a ARP e a GCS. No eixo horizontal, temos a distância e, no eixo vertical, o nível do sinal recebido.

No entanto, todos esses recursos e seus cabeçalhos de controle possuem um custo, e normalmente esse custo interfere na taxa de transferência de dados do enlace. Para determinar a taxa máxima de transmissão utilizamos o software *iPerf3* [32], que é uma ferramenta de medição para redes. Com o auxílio da ferramenta, executamos uma análise de sessenta segundos e outra de cento e vinte segundos. A análise ocorre de forma combinada entre os dois nós: durante o período de execução, um dos nós inunda o outro com pacotes e, baseado nas perdas, o software calcula a taxa máxima de dados transmitida. Na **figura 9**, podemos ver a saída no console do comando emitido. Os parâmetros passados no comando solicitam que um dos nós inunde o outro com mensagens do tipo UDP, com uma banda máxima de 300 kbps, durante um período de 60 segundos; além disso, ele solicita que os relatórios sejam impressos a cada 6 segundos e formatados em kilobits por segundo.

Durante os testes, capturamos os pacotes com o software wireshark [33] de duas maneiras diferentes. Uma das capturas foi feita no nível da interface de software, isto é, na interface de rede do sistema operacional; a captura neste ponto nos indica o

tamanho da banda útil para o uso de aplicações.

```
menezes@gcs:~/openthread$ iperf3 --format k -6 --udp -b 300k -i 6 -t 60 --reverse --client m4
Connecting to host m4, port 5201
Reverse mode, remote host m4 is sending
[ 5] local fd11:1111:1122:0:8f5b:b4cc:e6a:6ac2 port 51170 connected to fd11:1111:1122:0:fd2d
[ ID] Interval          Transfer          Bitrate          Jitter          Lost/Total Datagrams
[ 5] 0.00-6.00 sec      74.3 KBytes      101 Kbits/sec    358103.463 ms   87/150 (58%)
[ 5] 6.00-12.00 sec     72.0 KBytes      98.2 Kbits/sec   6994.475 ms    125/186 (67%)
[ 5] 12.00-18.00 sec    74.3 KBytes      101 Kbits/sec    127.237 ms     124/187 (66%)
[ 5] 18.00-24.00 sec    72.0 KBytes      98.2 Kbits/sec   10.962 ms     124/185 (67%)
[ 5] 24.00-30.00 sec    73.1 KBytes      99.9 Kbits/sec   12.901 ms     126/188 (67%)
[ 5] 30.00-36.00 sec    72.0 KBytes      98.3 Kbits/sec    5.543 ms     124/185 (67%)
[ 5] 36.00-42.00 sec    73.1 KBytes      99.9 Kbits/sec    6.587 ms     125/187 (67%)
[ 5] 42.00-48.00 sec    72.0 KBytes      98.3 Kbits/sec    6.864 ms     123/184 (67%)
[ 5] 48.00-54.00 sec    72.0 KBytes      98.2 Kbits/sec   12.517 ms     125/186 (67%)
[ 5] 54.00-60.00 sec    73.1 KBytes      99.9 Kbits/sec   10.746 ms     127/189 (67%)
-----
[ ID] Interval          Transfer          Bitrate          Jitter          Lost/Total Datagrams
[ 5] 0.00-60.24 sec    2.15 MBytes      300 Kbits/sec    0.000 ms       0/1870 (0%) sender
[ 5] 0.00-60.00 sec    728 KBytes       99.4 Kbits/sec   10.746 ms     1210/1827 (66%) receiver
```

Fig. 9 – Resultado da execução do software iPerf3.

A outra captura foi feita com um dispositivo nRF52840, carregado com um software de captura de pacotes 802.15.4; a captura neste ponto nos informa a capacidade máxima de transferências de dados na interface aérea. Na **figura 10**, apresentamos o resultado de ambas as capturas: no primeiro gráfico, a captura na interface aérea, com o valor de 195 kbps; e no segundo gráfico, a captura na interface de rede, com o valor de 100 kbps. Portanto, verificamos que o impacto da pilha de protocolos Thread na capacidade máxima de transmissão do protocolo 802.15.4 é de aproximadamente 51,28%.

A fim de estimar o valor médio de demandas que podem ser atendidas em cada cenário e o tempo médio de processamento, realizamos o segundo experimento. Nele simulamos o cálculo de posições das ARPs para estabelecer a rede de comunicação. Com base nos resultados do primeiro teste, configuramos o valor da distância máxima de enlace entre as aeronaves (δ) para 50 metros. Executamos simulações com demandas aleatoriamente posicionadas, variando a quantidade de demandas de 4 a 100, com passo incremental de 2 demandas.

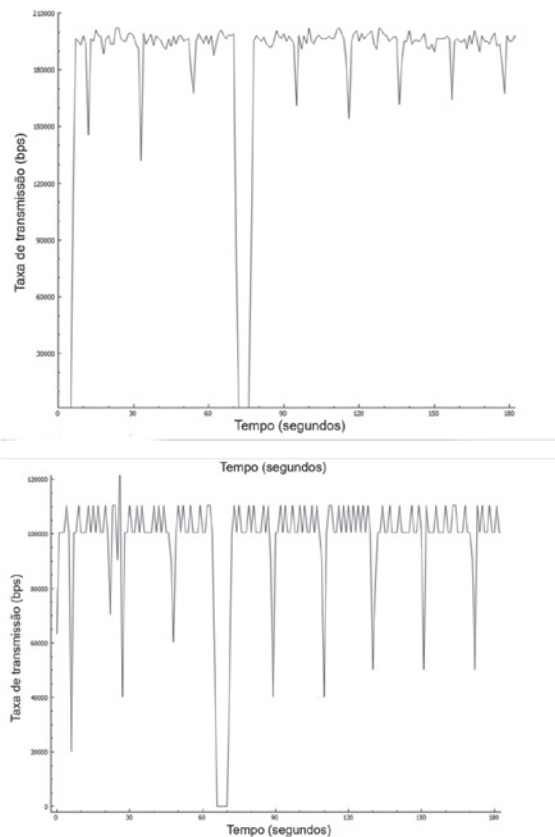


Fig. 10 – Teste usando a ferramenta iPerf3. Acima temos a taxa de dados transferida no nível de pacotes do protocolo 802.15.4 e, abaixo, a taxa de transferência na interface do sistema operacional.

Para cada quantidade de demandas, executamos 1000 simulações a fim de determinar a quantidade média de veículos necessários e o tempo de médio de execução do algoritmo. O experimento demonstrou que o cálculo das posições poderia ser realizado em tempo real e definiu o número médio de demandas, que podem ser atendidas com diferentes conjuntos de ARPs. A **figura 11** representa graficamente os resultados de quantas demandas podem ser atendidas (gráfico superior) e o tempo de execução do algoritmo (gráfico inferior).

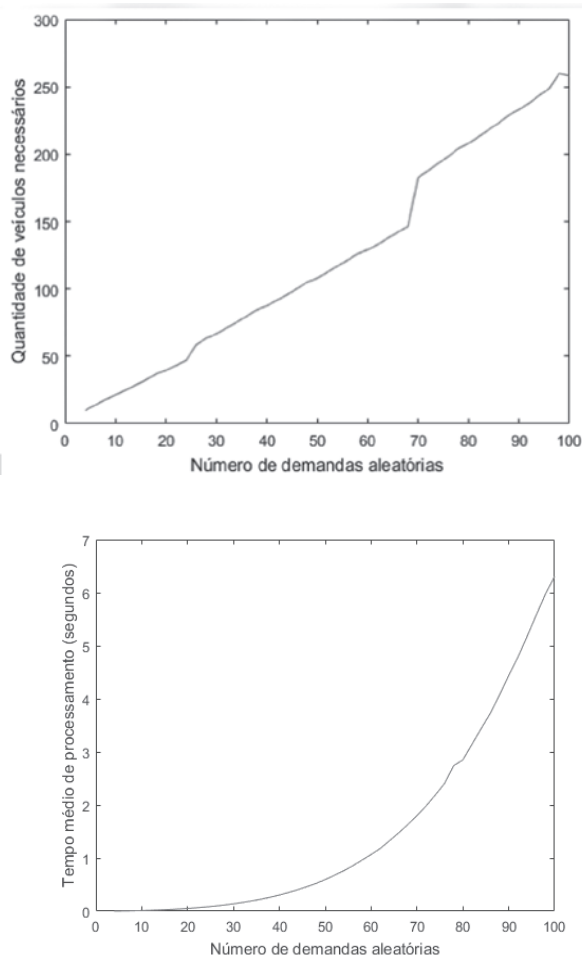


Fig. 11 – Cálculo de posição usando pontos de Fermat: O gráfico superior lista o número de veículos necessários (eixo

vertical) para atender a um determinado número de agentes terrestres (eixo horizontal). Observa-se o aumento linear da capacidade de serviço do sistema junto com o aumento do número de veículos disponíveis. O gráfico inferior mostra o tempo médio em segundos (eixo vertical) necessário para executar o cálculo da posição, considerando o número de agentes terrestres a serem atendidos (eixo horizontal).

Após a análise dos resultados do segundo experimento, extraímos os requisitos operacionais, resumidos na **tabela 2**. Conforme indicado, em média, para um conjunto de 50 ARPs, podemos atender até 24 demandas; para um conjunto de 150 ARPs, podemos atender até 68 demandas; e para um conjunto de 250 ARPs, podemos atender até 96 demandas. Apesar da curva de crescimento, vale ressaltar que o tempo máximo de processamento não ultrapassou 7 segundos. Tal resultado é consequência da simplicidade do algoritmo desenvolvido e, principalmente, da baixa quantidade de veículos disponíveis para uso nos cenários operacionais apresentados. O experimento também apurou que a economia de veículos necessários varia de 26, 30% a 38,82%, em relação ao método de planejamento direto de trajetórias; onde cada demanda é atendido por uma subárvore direta de aeronaves.

Tab. 2 – Estimativas médias de planejamento de trajetória com ponto de Fermat.

Resultados	Cenário 1	Cenário 2	Cenário 3
Número médio de demandas atendidas	24	68	96
Duração média da execução do algoritmo (segundos)	0,0787	1,6326	5,5796
Economia de veículo em relação ao planejamento direto (%)	26,3014	33,7374	38,8220

No terceiro experimento, o algoritmo foi

integrado na GCS desenvolvida por Silva [34] no intuito de viabilizar o ensaio em voo do sistema completo. De posse dessa implementação, pudemos apresentar uma prova de conceito inicial e reutilizar um componente já testado em outros trabalhos [30-31, 34]. Na **figura 12**, podemos observar um centro C2, três demandas dadas como entrada para o algoritmo de cálculo de posições e onze ARPs. Note que há uma árvore de ARPs conectando todas as demandas à sua raiz, o centro C2.

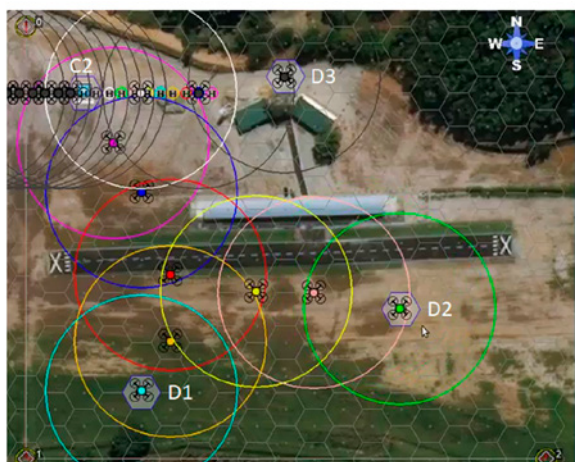


Fig. 12 - A simulação mostra três demandas conectadas ao centro C2 através de uma rede formada por 11 ARPs.

6. Conclusão

Veículos aéreos não tripulados têm um potencial considerável para alavancar operações de recuperação de desastres. Além da aplicação descrita neste artigo, as ARPs podem ajudar a monitorar as atividades das equipes em solo, avaliar as áreas afetadas e capturar dados para reconstrução 3D das áreas de risco.

Este artigo descreveu uma solução para empregar um conjunto de ARPs para formar uma rede sobre uma área afetada por desastres, conectando todos os agentes em solo ao centro de comando e controle. Os resultados dos experimentos mostram que essa rede é viável com o dispositivos do tipo Thread, apesar da limitação de alcance já esperada do dispositivo nRF52840-dongle.

Outra consideração interessante do sistema é o mecanismo de continuidade. Atualmente, usamos o sistema de substituição de ARP para tarefas contínuas [35].

Agradecimentos

Os autores agradecem a Ricardo Maroquio Bernardo por compartilhar seu projeto de quadricóptero e por ajudar na montagem das ARPs utilizadas nos experimentos.

Referências bibliográficas

- [1] Saska, M.; Chudoba, J.; Precil, L.; Thomas, J.; Loianno, G.; Tresnak, A.; Vonasek, V.; Kumar, V. Autonomous deployment of swarms of micro-aerial vehicles in cooperative surveillance. 2014 International Conference on Unmanned Aircraft Systems, ICUAS 2014 - Conference Proceedings, v. 1, p. 584–595, 2014.
- [2] Šulák, V.; Kotuliak, I.; Čičák, P. Search using a swarm of unmanned aerial vehicles. 2017 15th International Conference on Emerging eLearning Technologies and Applications (ICETA), v. 1, p. 1–6, Oct 2017.
- [3] Kato, A.; Wakabayashi, H.; Hayakawa, Y.; Bradford, M.; Watanabe, M.; Yamaguchi, Y. Tropical forest disaster monitoring with multi-scale sensors from terrestrial laser, uav, to satellite radar. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). 2017. p. 2883–2886.

- [4] Hartman, J.; Barzilov, A.; Novikov, I. Remote sensing of neutron and gamma radiation using aerial unmanned autonomous system. 2015 IEEE Nuclear Science Symposium and Medical Imaging Conference, NSS/MIC 2015, v. 1, p. 1–4, 2016.
- [5] Micconi, G.; Aleotti, J.; Caselli, S.; Benassi, G.; Zambelli, N.; Zappettini, A. Haptic guided uav for detection of radiation sources in outdoor environments. In: 2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS). Mexico: IEEE, 2015. p. 265–271.
- [6] Yu, H.; Yang, W.; Zhang, H.; He, W. A uav-based crack inspection system for concrete bridge monitoring. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). 2017. p. 3305–3308.
- [7] Ikeda, T.; Yasui, S.; Fujihara, M.; Ohara, K.; Ashizawa, S.; Ichikawa, A.; Okino, A.; Oomichi, T.; Fukuda, T. Wall contact by octo-rotor uav with one dof manipulator for bridge inspection. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2017. p. 5122–5127.
- [8] Li, Z.; Mu, S.; Li, J.; Wang, W.; Liu, Y. Transmission line intelligent inspection central control and mass data processing system and application based on uav. In: 2016 4th International Conference on Applied Robotics for the Power Industry (CARPI). 2016. p. 1–5.
- [9] Shukla, A.; Karki, H. Teleoperation of a uav by a 6-dof parallel manipulator with hybrid mapping for inspection of industrial structures. In: 2017 17th International Conference on Control, Automation and Systems (ICCAS). 2017. p. 883–888.
- [10] Kumar, S. A.; Vanualailai, J. A Lagrangian UAV swarm formation suitable for monitoring exclusive economic zone and for search and rescue. In: 2017 IEEE Conference on Control Technology and Applications (CCTA). [S.l.]: IEEE, 2017. p. 1874–1879. ISBN 978-1-5090-2182-6.
- [11] Paula, P. de S.; Castro, M. F. de; Paillard, G. A. L.; Sarmiento, W. W. F. A swarm solution for a cooperative and self-organized team of uavs to search targets. 2016 8th Euro American Conference on Telematics and Information Systems (EATIS), v. 1, p. 1–8, April 2016.
- [12] Ministério da Defesa, EMCFA. Doutrina para o sistema militar de comando e controle, 3a Edição. 2015. Disponível em: <www.defesa.gov.br>. Acesso em: 26 de fevereiro de 2018.
- [13] Dong, S. Y. Optimization of olsr routing protocol in uav ad hoc network. In: 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWA-MTIP). 2016. p. 90–94.
- [14] Hussien, H. R.; Choi, S.; Park, J.; Kim, J. Performance analysis of manet routing protocols for uav communications. In: 2018 10th International Conference on Ubiquitous and Future Networks (ICU-FN). 2018. p. 70–72. ISSN 2165-8536.
- [15] Mahmud, I.; Cho, Y. Adaptive hello interval in fanet routing protocols for green uavs. IEEE Access, v. 7, p. 63004–63015, 2019. ISSN 2169-3536.
- [16] Zhang, Q.; Jiang, M.; Feng, Z.; Li, W.; Zhang, W.; Pan, M. Iot enabled uav: Network architecture and routing algorithm. IEEE Internet of Things Journal, v. 6, n. 2, p.3727–3742, April 2019. ISSN 2327-4662.
- [17] Ruan, L.; Wang, J.; Chen, J.; Xu, Y.; Yang, Y.; Jiang, H.; Zhang, Y.; Xu, Y. Energy-efficient multi-uav coverage deployment in uav networks: A game-theoretic framework. China Communications, v. 15, n. 10, p. 194–209, Oct 2018. ISSN 1673-5447.
- [18] Li, X.; Xu, X. Uav comprehensive coverage to users in urban environment. In: 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS). 2018. p. 503–506. ISSN 2327-0594.
- [19] Urosevic, U.; Veljovic, Z.; Pejanovic-Djurisic, M. Improving cellular coverage through uavs. In: 2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC). 2017. p. 68–73. ISSN 1882-5621.

- [20] Li, B.; Chen, C.; Zhang, R.; Jiang, H.; Guo, X. The energy-efficient uav-based bs coverage in air-to-ground communications. In: 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM). 2018. p. 578–581. ISSN 2151-870X.
- [21] Gupta, L.; Jain, R.; Vaszkun, G. Survey of important issues in uav communication networks. IEEE Communications Surveys Tutorials, v. 18, n. 2, p. 1123–1152, Second quarter 2016. ISSN 1553-877X.
- [22] Abrajano, G.; Favila, C.; Luo, C. Y.; Trono, E.; Lagazo, D.; Sevilla, B.; Honrado, J.; Solpico, D.; Yu, J.; Chua, K.; Mamaradlo, J.; Jose, C.; Yao, C. J.; dela Cruz, J.; Ancheta, E.; Domingo, A.; Ong, J.; Datuin, J.; Yasumoto, K.; Libatique, N.; Tangonan, G. Demonstrations of post-disaster resilient communications and decision-support platform with uavs, ground teams and vehicles using delay-tolerant information networks on sub-ghz frequencies. In: 2017 IEEE Global Humanitarian Technology Conference (GHTC). 2017. p. 1–8.
- [23] Nest Labs. OpenThread. 2019. Disponível em: <<https://openthread.io/>>. Acesso em: 19 de janeiro de 2019.
- [24] Ardupilot Team. Mission planner. 2019. Disponível em: <<https://ardupilot.org/planner/>>. Acesso em: 19 de janeiro de 2019.
- [25] Ardupilot Team. Firmware arducopter. 2019. Disponível em: <<https://ardupilot.org/copter/>>. Acesso em: 19 de janeiro de 2019.
- [26] Ardupilot Team. Controladora de voo kakute F7 AIO. 2019. Disponível em: <<https://ardupilot.org/copter/docs/common-holybro-kakutef7aio.html>>. Acesso em: 19 de janeiro de 2019.
- [27] Friendly ARM. Computador Nano Pi Neo Air. 2019. Disponível em: <http://wiki.friendlyarm.com/wiki/index.php/NanoPi_NEO_Air>. Acesso em: 19 de janeiro de 2019.
- [28] Wpantund Team. wpantund software. 2019. Disponível em: <<https://github.com/openthread/wpantund>>. Acesso em: 22 de maio de 2019
- [29] DARPA. OFFensive Swarm Enabled Tactics (OFFSET). 2017. Disponível em: <<https://www.fbo.gov/utills/view?id=015bda98560d82d7568501376efaebdf>>. Acesso em: 02 de dezembro de 2017.
- [30] Silva, L. C. B. da; Bernardo, R. M.; Oliveira, H. A. de; Rosa, P. F. F. Multi-uav agent-based coordination for persistent surveillance with dynamic priorities. In: 2017 International Conference on Military Technologies (ICMT), 2017. p.765–771.
- [31] Silva, L. C. B. da; Bernardo, R. M.; Oliveira, H. A. de; Rosa, P. F. F. Unmanned aircraft system coordination for persistent surveillance with different priorities. In: 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), 2017. p. 1153–1158. ISSN 2163-5145
- [32] ESnet - Lawrence Berkeley National Laboratory. iperf3. 2019. Disponível em: <<https://github.com/esnet/iperf3>>. Acesso em: 22 de maio de 2019.
- [33] Combs, G. Wireshark. 2019. Disponível em: <<https://www.wireshark.org/>>. Acesso em: 22 de maio de 2019.
- [34] Silva, L. C. B. Coordenação de multi-agentes robóticos para controle de VANT sem monitoramento persistente por cobertura de área com prioridades dinâmicas. Tese de doutorado, Instituto Militar de Engenharia, Rio de Janeiro, RJ, Brasil, 2018.
- [35] Oliveira, N.; Moreira, E. M.; Rosa, P. F. F. Particle swarm optimization algorithm implementation for multiple drones control in continuous task simulation. In: 16th IEEE Latin American Robotics Symposium (SBR-LARS 2019). Rio Grande, Brazil: IEEE, 2019.